

Management of the Migration in Virtual Resource

Yanna Yan, Liang Shao

Department of humanity and information, Zhejiang College of Construction

yyanna425@sina.com; shaoliang.hz@gmail.com

Abstract—Large-scale data centers, the workload in each node changes dynamically over time, causing the imbalance of workload, so the workload in some machines may exceeds the threshold, which not only can't guarantee the quality of service requirements (QoS), may also waste resources. This paper designs a resource manager based on migration cost, resource utilization and the number of machines, which develops relevant migration policy for virtual machine migration. The manager is divided into two stages. The initial phase will decide the virtual machines (VM) which should be migrated. The goal is to meet QoS, maximize resource utilization and minimize migration overhead. The second phase develops migration program to resolve the issue of where to move to. The target is to improve resource utilization and reduce the number of machines. Simulation results verify the superiority and rationality of this resource manager.

Keywords—Virtual machine; dynamic migration; quality of service requirements; resource utilization

I. INTRODUCTION

An urgent need to focus on reduced total cost of ownership (TCO) for enterprise computing infrastructure, so virtualization has become a key requirement for the enterprise. Server virtualization that allows applications to be hosted inside virtual servers instead of physical ones has become more and more important. A challenge that comes with virtualization is how to meeting application-level quality of service (QoS) goals and how to manage resource to increasing resource utilization in a consolidated environment as application resource needs differ.

In large-scale data center or cluster environment, the change of workload is random and uncertain. In order to ensure the QoS of workload, it is essential to effective manage the resource of data center. Using virtualization technology[1] can lower the manage complexity, reduce the number of physical machines, save resources.

Currently, Virtualization technology can bring great convenience, and the biggest advantage is its ability to help achieve the dynamic resources allocation in data center. Because workloads change dynamically over time, a virtual machine at a certain moment may not be active, but may be active in next moment, resulting in overload of physical machines (PM). At this time, we need to move out some VMs from the PM to ensure the QoS of workload. Live migration of virtualization[2] can be used to migrating the superfluous VMs of overload PM. Live migration is that the virtual machine maintains running, and at the same time, the VM is migrated from one computer to another computer, then which would run in the purpose computer. It can achieve the dynamic allocation of physical resources. Live migration of virtual machines has been a powerful tool to facilitate system maintenance, load balancing, fault tolerance, and power-saving, especially in clusters or data centers. Although pre-copy is a predominantly

used approach in the state of the art, it is difficult to provide quick migration with low network overhead, due to a great amount of transferred data during migration, leading to large performance degradation of virtual machine services.

At present, the migration problem in cluster and multi-machine resource management has become a hot spot [3-6]. While virtual machine (VM) migration is allowing data centers to rebalance workloads across physical machines, the promise of a maximally utilized infrastructure is yet to be realized. Part of the challenge is due to the inherent dependencies between VMs comprising a multi-tier application, which introduce complex load interactions between the underlying physical servers. If overload, we should go to migration management, but which VM is the most suitable to migrate? Different objectives ask different VMs to migrate. How to develop the migration program? Migrate to Where that we can solve the overload and ensure the QoS requirements, and at the same time, achieve other objectives of the system.

In this paper, multi-machine resource management is divided into two phases. This paper designs a resource manager based on QoS goal, migration cost, resource utilization and the number of machines, which develops relevant migration policy for virtual machine migration. In the first phase, the goal is to determine which VM should be migrate based on meeting QoS, maximize resource utilization and minimize migration overhead. The second stage solves the problem where virtual machine should be to migrate, with the aim of least number of physical machines and the largest resource utilization. This is a special bin-packing issue, but also a NP hard problem. This paper designs related heuristic algorithm for the two phases, respectively. Through simulation experiments, the manager has got a better result in resource utilization, the number of physical machines and energy consumption of systems.

The remainder of the paper is organized as follows. In Section 2, we review related work and distinguish our own contributions from the existing literature. In Section 3, we introduce resource manager model which present the overall resource manager model framework and objectives. Then, describe the design of the total two-stage and related algorithms. In Section 4, present and discuss some simulation experimental results of the resource manger model and the related algorithms, and compared with the FFD algorithm. Finally, in Section 5, we offer conclusions.

II. RELATED WORK

Virtual machine migration can be formulated to the multi-dimensional bin-packing problem [7]. As we all know, bin-packing problem is a NP hard problem, so far do not have the optimal solution. To address this issue, a number of articles [8, 9] proposed polynomial time approximate algorithms, but the approximate rate is low. Tracy [10] gives a limit load allocation

algorithm which has the lowest migration constraint, but this algorithm cannot allow more loads assigned to a machine, and also assumed that the dimension and the box is fixed. In a virtual environment, virtual machine utilization is changing over time, so the static allocation method is inappropriate. In order to accurately represent the changes of load, paper [11] proposed random load balancing, setting the probability limits of resources. In the bin-packing problem, using a random or a static solution largely depends on whether the initial allocation is close to the optimal.

Now, resource management in large-scale virtual environment has become a research focus [12]. Chris and others [3] present a VM allocation program, which improved the resource utilization, reduced the overload situation, but there was no discussion on overload. Paper [4] proposes a manage algorithm for the dynamic allocation of VMs, and then performs dynamic migration of VMs using the FFD heuristic algorithm based on ensuring the quality of service requirements, however, Norman and others do not take into account the migration cost and energy consumption of system. Literature [5] considers the migration overhead, formulates the problem as an integral constraint satisfaction problem (CSP) problem, and then presents the allocation program, but it should be in the grid or high-performance computing environment.

Timothy [6] presents a sandpiper systems, this system also using the FFD heuristic algorithm to complete the configuration of the migration program, although it presents a comprehensive analysis, prediction and allocation for network, memory, and CPU resources, but has failed to consider the optimize results of resources allocation and the number of physical machines which are used in this system. Jong-Geun P et al. [13] introduce an automated strategy for virtual machine migration in self-managing virtualized server environments. Hai Jin et al. [14] propose a novel memory-compression- that uses memory compression to provide virtual machine migration. It is useful for balancing the performance and the cost of virtual machine memory migration.

We know that simply moving an overloaded VM to a (random) underloaded physical machine can inadvertently overload the network. So Shrivastava et al. [15] introduce AppAware—a novel, computationally efficient scheme for incorporating inter-VM dependencies and the underlying network topology into VM migration decisions, which decreases network traffic. However, compared to a well known alternative VM migration method, it is not application-aware. For the total migration time, pre-copy approach can balance the downtime and total migration time of contradictions. Fei Ma et al. [16] add a bitmap page to improve the pre-copy approach., which marks those frequently updated pages. That approach reduces the total migration time.

In this paper, a two-stage model and algorithms are present to study the resources management. This paper designs a resource manager based on QoS goal, migration cost and system power consumption, which optimizes the resources allocation program. The simulation result behind the article verifies the effectiveness and superiority of the algorithms we design for the resource manager.

III. RESOURCE ALLOCATION MODEL

In generally, the architecture of virtual machine system can be shown in Figure 1. In the architecture, multiple virtual machines (VMs) share the same “physical machine”. Various resources allocated to VMs by virtual machine monitor (VMM) which is responsible for the management of resources. Virtual machine monitor allocated physical resources to the virtual machine based on static or dynamic allocation strategy. Multiple virtual machines will contend for these physical resources when tasks are allocated to these virtual machines because of the same physical resources they shared in virtual machine system.

Static resources allocation determines the allocation of resources when the virtual machine created. Dynamic resource allocation regulates the resources allocation value to insure the application to meet the QoS goal based on the different resource requirements of VMs, while the resources utilization rate can also achieve a certain level. The QoS goal will change according to different application environments. It can be response time, throughput, delay time, computing speed, etc. Live migration can achieve the dynamic allocation of physical resources.

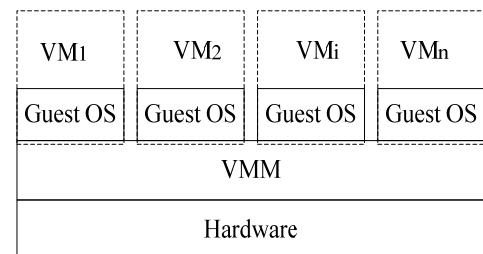


Figure 1. Architecture of virtual machine system

A. Model Overview

Multi-machine resource management model in virtual environment is shown in Figure 2. In the first stage, we determined VMs which need to migrate; the goal is maximize resource utilization and minimize the migration overhead. And then, in the second phase, global control for VMs which need migration, resolve the problem where to migrate to. The migration program should meet the quality of service requirements, while at the same time the number of new use physical machines is least.

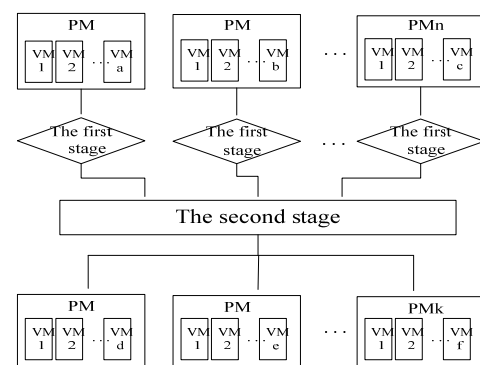


Figure 2. Resource allocation control model.

This paper set the CPU utilization ratio as the goal, and assuming that when the CPU utilization ratio reaches a certain level, the QoS goal is met. The related assumptions and parameters are defined as follows:

- Suppose it has sufficient idle machines except the running physical machines, which means the total number of physical machines is uncertain.
- Resource requirements of each virtual machine will not exceed the sum useful resources of the machine.
- Assuming that we have got the predictive value of resource requirement for each virtual machine in various physical machines.

The related parameters definition is shown in table 1.

TABLE I. PARAMETERS DEFINE

parameter	define
n	The number of using PM
m	The largest number of VM which live in a PM
m_i	The number of VM living in a_i , no more than the largest support number of PM, $m_i \leq m$
L	The number of new using PM
w_{ij}	The resource requirement of VM j in a_i
u_{ij}	The allocation resource for virtual machine V_{ij} in a_i
M_i	The threshold
$\tilde{M}_i$	The resource allocation error factor of V_{ij}
μ	The total resource of a_i , which is the threshold of a_i too
e_{ij}	The utilization target of a_i
C_i	
R_i	

In the resource manager, we set up the corresponding objectives at every stage. The overall objective is to ensure the QoS requirements while reducing the use of new machines. The objective function of resource allocation manager model is as follows:

$$\begin{aligned} & \begin{cases} \min \sum_{j=1}^{m_i} \left(\frac{v_{ij}}{u_{ij}} - R_i \right)^2 \\ \min \sum_{\tilde{M}_k \in New} k \end{cases} \\ \text{s.t. } & 0 \leq \sum_{j=1}^{m_i} u_{ij} \leq C_i, \quad i=1,2,\dots,n, \\ & j=1,2,\dots,m_i, \quad k=1,2,\dots \end{aligned}$$

The objective function of above model is a multi-objective optimization problem. Aim is to minimize the deviation between the target utilization and the actual utilization of each virtual machine and the number of new use machines.

B. The First Phase

The finder is important in first stage. It will find the VMs which need to move in the overload PM. We set the relevant algorithms inside the finder, which can be used to find the VMs which need to migrate, and its goal is to meeting QoS requirements, reducing the migration costs. We assume that the fewer number of migrations, less migration costs, that is, the migration cost is proportional to the migration number. Determining which virtual machine need to migrate is a NP hard problem, cannot get the optimal solution, therefore, this paper designed a heuristic algorithm: search the virtual machine algorithm (SVMA). It can find the virtual machines which need for migration by the smallest migration overhead.

Specific algorithm is as follows:

Input: The resource requirements w_{ij} of all VMs on a machine and $C_i, j=1,2,\dots,m_i, i=1,2,\dots,n$

Output: The VMs set S_i which need to move out, $S_i = \{v_{ij}\}$ and the resource allocations of remaining VMs.

Step1: If $\sum_{j=1}^m w_{ij} \leq C_i, i=1,2,\dots,n$, output:

$$u_{ij} = \frac{w_{ij}}{R_i} \cdot e_{ij}, \text{ the number of virtual machine which needs to move out is zero. Else go to Step2.}$$

Step2: Put the virtual machine by w_{ij} ascending order.

If $\sum_{v_{ij} \in S_r} w_{ij} < C_i$, add v_{ij} to the set S_r one by one, until put into the V_{ik} , the sum resource requirements over the threshold, stop, turn Step3.

Step3: If $\sum_{j=1}^{k-2} w_{ij} + w_{ik} < C_i$, and $\frac{C_i}{\sum_{j=1}^{k-2} w_{ij} + w_{ik}} < \mu$, add V_{ik} into S_r and get out V_{ik-1} from S_r .

Otherwise, keep unchanged. Repeat Step3 for every VM until the final one.

Step4: Putout $u_{ij} = w_{ij} / R_i$ and S_i , $v_{ij} \in S_r$, $S_i = \overline{S_r}$.

Use the SVMA algorithm can find the migration VMs quickly based on the least migration cost. At the same time, not only the virtual machines which remaining in the physical machine can meet the QoS goal, but also the physical machine can have high resource utilization.

C. The Second Phase

The objectives of the migration program are the least number of new use physical machines and the minimum energy consumption. We assume that the more physical machines use of, the greater the energy consumption, that is, the energy consumption of machines is directly proportional to the number of physical machines. However, the resource of physical machine are limited, the number of VMs which live in machine is also limited. Therefore, the objective function model of this stage as follows:

$$\begin{aligned} \min \quad & \sum_{\tilde{M}_i \in \text{New}} i \\ \text{s.t.} \quad & \begin{cases} 0 \leq \sum_{j=1}^{m_i} u_{ij} \leq C_i \\ m_i \leq m \end{cases} \\ & i = 1, 2, \dots, j = 1, 2, \dots, m_i \end{aligned}$$

At this stage, we will develop the migration program and rules, through the migration, VMs will get the resources they need, and meet the QoS requirement. To develop the migration program is a special bin-packing problem, which is a NP hard problem. This paper presents a fast heuristic algorithm (Migration) to use to solve the problem.

Specific algorithm is as follows:

Input: $S = \{S_i\}$, $i = 1, 2, \dots, n$.

Output: The number of new use machines L and resource allocations of VMs on all machines.

Step1: Put the virtual machine in S by w_{ij} descending order, and get the set $V = \{v_{ij}^1, \dots, v_{ij}^x\}$, allocate the resource to VMs which in V one by one. Initial set up $t=1, l=1$

Step2: Open a new machine, migrate v_{ij}^t to the new machine M_l , which placing in the last of all open machines, the number parameter of M_l is $Z^l=1$, the VM set of M_l is V^l . Setting $v_{ij}^t \in V^l$, $z = l$, $l = 1$ and $t = t + 1$, turn to Step3.

Step3: For the v_{ij}^t on S , make the following judgment:

If $Z^l + 1 > m_l$, turn to Step4.

If $Z^l + 1 \leq m_l$, and $\sum_{v_{ij} \in V^l} w_{ij} + w_{ij}^t \leq C_l$, set $v_{ij}^t \in V^l$, $Z^l = Z^l + 1$, $t = t + 1$, $l = l + 1$, turn to Step3.

Else if $\sum_{v_{ij} \in V^l} w_{ij} + w_{ij}^t > C_l$, and $l < z$, set $l = l + 1$, turn to Step3.

If $\sum_{v_{ij} \in V^l} w_{ij} + w_{ij}^t > C_l$, and $l = z$, set $l = l + 1$, turn to Step2.

Step4: Remove the machine M_l from the queue, and the remaining machines keep the original order, followed by label $M_1, M_2, \dots, M_{z-1}$, if $t = X$, turn to Step5, else set $z = z - 1$, $t = t + 1$ and $l = 1$, return to Step3.

Step5: Allocating resource for all machines in accordance

$u_{ij} = \frac{w_{ij}}{R_i} \cdot e_{ij}$, with R_i , output the allocations u_{ij} of all VMs on each machine.

IV. SIMULATION AND PERFORMANCE ANALYSIS

This paper uses C++ to do the algorithms (SVMAM: SVMA and Migration) simulation experiments, and then compared the simulation results of SVMAM and FFD algorithm. The number of VMs on each physical machine meets a random distribution function, and the resources demand of load on virtual machines is also random.

Figure 3 presents the change of ratio (NPM/TPM) which is the number of new physical machines (NPM) divide to the total number of used physical machine (TPM) follows the increasing number of physical machines. From the graph, we can seen, the NPM/TPM value of SVMAM algorithm is far below the FFD algorithm. That means if the total number of physical machines is same, the NPM value of SVMAM algorithm is far less than the FFD algorithm, so the energy consumption of SVMAM is also less. The average NPM / TPM ratio of SVMAM algorithm is about 60% lower than FFD algorithm, the maximum of 68%.

The change of NPM / TPM ratio from SVMAM algorithm is more gentle and stability, almost at about 0.08, the change range is in 0.08 ± 0.01 , that is to say, with the number of

physical machines from 50 to 500, the increase number of new physical machines relative to the total number of previously used physical machines is very small, and the ratio is about 0.08. However the change of NPM / TPM ratio from FFD algorithm is relatively large. This indicates that the SVMAM algorithm proposed in this paper is more reasonable and beneficial than FFD in reduce the number of new use physical machines. At the same time, the resource allocation manager model is stable.

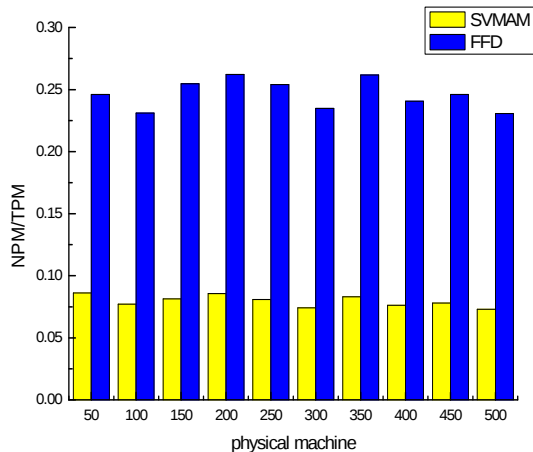


Figure 3. NPM/TPM value of SVMAM vs. FFD

Figure 4 presents the change of ratio which is the total resource of VMs which need to migrate divide to the number of physical machines follows the increasing number of physical machines. By the map, we can see that when the number of machines from 50 to 500, the ratio from SVMAM algorithm is lower than FFD. That is because when designing the algorithm SVMA, its goal is improving the resource utilization rate, which means using the greatest machine resource is the target. And for SVMAM algorithm, the total resource requirement of VMs which need to migrate relative to the total resources of physical machine is very small, and the ratio is about 0.07, which shows the migration cost is small.

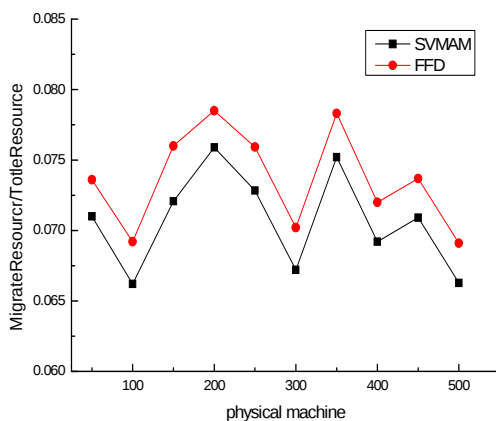


Figure 4. Resource of migrate VM required /Total resource of SVMAM vs. FFD

Figure 5 illustrates the changes of average resource utilization rate of new using machine. In the experiment, we set the target utilization rate of 80%. Clearly, the resource utilization from SVMAM algorithm is closer to the target, almost in error range which we set of. That is, the SVMAM algorithm can guarantee the QoS requirement, and the change is relative stable. The resource utilization from FFD algorithm is lower, has a certain distance from the target value.

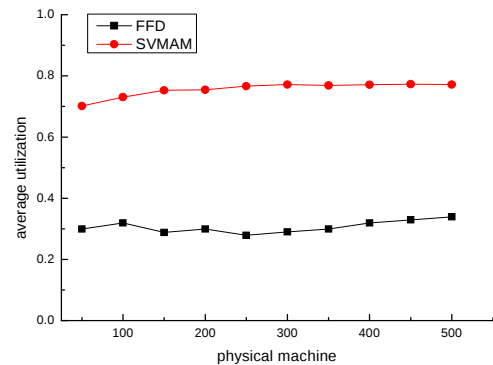


Figure 5. Average resource utilization of SVMAM vs. FFD

V. CONCLUSIONS

This paper presents a resource allocation manager based on some constraints. In the initial stage, it determines VMs which need to migrate under the constraints of meeting the QoS goals, maximizing resource utilization and minimizing migration overhead. For this multi-constraint optimization problem, an improved heuristic algorithm SVMA is present. The goal of second phase is least number of physical machines and maximum resource utilization, then develops and designs the migration program and Migration algorithm accordingly. The contrast results of simulation experiments show that compared with FFD algorithm, the SVMAM algorithm of this Manager has get good results and do the greatly improved on resource utilization, the number of using physical machines and systems energy consumption. So the method proposed in this paper is superior and rational.

REFERENCES

- [1] Paul, B., et al., Xen and the art of virtualization, in Proceedings of the nineteenth ACM symposium on Operating systems principles. 2003, ACM: Bolton Landing, NY, USA.
- [2] Robert, B., et al., Live wide-area migration of virtual machines including local persistent state, in Proceedings of the 3rd international conference on Virtual execution environments. 2007, ACM: San Diego, California, USA.
- [3] Hyser, C., et al., Autonomic Virtual Machine Placement in the Data Center, in HP Labs Technical Report HPL-2007-189. 2007.
- [4] Bobroff, N., A. Kochut, and K. Beaty. Dynamic Placement of Virtual Machines for Managing SLA Violations. in Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on. 2007.
- [5] Fabien, H., et al., Entropy: a consolidation manager for clusters, in Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments. 2009, ACM: Washington, DC, USA.

- [6] Timothy, W., et al. Black-box and gray-box strategies for virtual machine migration. in Proceedings the Fourth Symposium on Networked Systems Design and Implementation (NSDI). 2007.
- [7] Srinivas, K. and K. Samir, Algorithms for non-uniform size data placement on parallel disks. J. Algorithms, 2006. 60(2): p. 144-167.
- [8] Shachnai, H. and T. Tamir, Noah's bagel-some combinatorial aspects, in International Conference on FUN with algorithms (FUN). 1998: Elba. p. 65—78.
- [9] Chandra, C. and K. Sanjeev, On multi-dimensional packing problems, in Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms. 1999, Society for Industrial and Applied Mathematics: Baltimore, Maryland, United States.
- [10] Tracy, K., S. Baruch, and S. Maxim, Minimizing migrations in fair multiprocessor scheduling of persistent tasks, in Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms. 2004, Society for Industrial and Applied Mathematics: New Orleans, Louisiana.
- [11] Goel, A. and P. Indyk. Stochastic load balancing and related problems. in Foundations of Computer Science, 1999. 40th Annual Symposium on. 1999.
- [12] L. Ljung, "System Identification: Theory for the User (2nd Edition)". 1999: Prentice Hall.
- [13] Jong-Geun P et al. Virtual machine migration in self-managing virtualized server environments. In Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on. 2009, p. 2077,
- [14] Hai Jin et al. Live virtual machine migration with adaptive, memory compression. In Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on, 2009
- [15] Shrivastava, Vivek Zerkos, et al, Application-aware virtual machine migration in data centers. in INFOCOM, 2011 Proceedings IEEE. 2011, p. 66.
- [16] Fei Ma et al., Live virtual machine migration based on improved pre-copy approach. in Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on. 2010, p. 230